# DAQuA: Difficulty Aware Question Answering

**Dhruv Agarwal** and **Nishant Raj** and **Akshay Sharma**
College of Information and Computer Sciences
University of Massachusetts Amherst

## Abstract

State-of-the-art question answering systems work in two phases: a term- or dense-based retriever model is first used to fetch passages or documents based on coarse relevance with the question, followed by fine-grained span extraction using a reader model. While this approach is effective, it is agnostic to the difficulty of the question and incurs identical latency for questions of varying levels of difficulty as defined by the number of hops. Our work aims to design a low-latency system that is able to answer both single- and multi-hop questions with adaptive compute requirements. We hypothesize that phrase-level retrieval provides sufficient information to both answer simple questions as well as build valid evidence chains for multi-hop questions, obviating the need for an expensive reader module. We propose an adaptive multi-hop question-answering system using phrase retrieval only.

## 1 Introduction

### 1.1 Task Description

Our work is primarily based on Question Answering sub-domain of natural language processing. Based on an extensive literature review, we found out that only a handful of works focus on designing a pipeline that can adapt to the difficulty of the questions. We aim to address the pitfalls in the existing approach & develop a system that can identify as well as adapt to the complexity associated with the question in the open domain question answering environment. We believe that phrase level retrievals instead of the passage level retrievers might be helpful in our setting and thus we try to address two major research questions at hand:

- **RQ1**: Can we develop an approach that is able to handle both natural questions in single hop settings & multi-hop questions where connections over multiple evidence chains are required ?
- **RQ2**: Dense Phrases representation can be helpful in guiding towards the answer phrase directly instead of a two level retriever reader approach. We aim to address the question whether these dense phrase representations are sufficient for the architecture that we plan to design above.

### 1.2 Motivation and Limitations of Existing Work

A significant amount of work has been done in Open Domain Question Answering (ODQA) field with varying levels of success. While the initial approaches in the retriever-reader architecture used sparse vector space models like TF-IDF, BM25 (Robertson and Zaragoza, 2009) etc., the focus gradually shifted to more accurate dense retrievers in work like Dense Passage Retrieval (DPR) (Karpukhin et al., 2020), Retrieval Augmented Language Model Pre-Training (REALM) (Guu et al., 2020) etc. While these approaches are remarkably efficient in the information retrieval, they increase the computational costs by orders of magnitude greater than prior sparse vector approaches. Some works like Contextualized Late Interaction Mechanism Over BERT (ColBERT) (Khattab and Zaharia, 2020) propose the use of the late interaction mechanisms to tackle the issue outlined above. Phrase Indexed Question Answering (PIQA) (Seo et al., 2018) paper highlights a research direction where they leverage phrases as answer documents directly. This approach had the added benefit that all the phrases in the documents could be pre-computed and indexed offline for efficient retrievals. Dense Phrases (Lee et al., 2020) paper builds on this idea however they were the first to highlight the fact that dense phrase representations alone could achieve

much stronger performance in open-domain question answering.

However, these works only address the questions that involve searches that can be answered in single hops and falters when multiple evidence chains for reasoning is involved. For example, consider a question: *What is the capital of the country where Steve Jobs went in search of spiritual enlightenment?*. Here, one might not find the answer in a single wikipedia document and might have to connect evidence chains from separate documents to arrive at the final answers. In the above example, this could involve three hops: 1) Steve Jobs went to see Neem Karoli Baba at his Kanchi ashram in Uttarakhand 2) Uttarakhand is a state in India 3) New Delhi is both a union territory and capital of India to arrive at the final answer which is "New Delhi".

Many existing works have tried to address the issue of multi-hop question answering. **Hotpot QA** (Yang et al., 2018) is on such dataset hosted as a challenge. In multi-hop settings, the search space continuously grows with each retrieval step. Existing works that use hyperlinks or an entity linking mechanisms are not very scalable and can't be used in other domains. New line of research aims to devise an iterative framework that takes evidences from successive retrievals and use that to augment the question in order to make the retrieval framework more elegant and efficient.

We propose an adaptive dense phrase retrieval based framework that can answer any question in an open domain settings. We hypothesize that our framework would be able to adapt to any level of difficulty. Our work takes an iterative approach in order to deal with multiple hops based on the difficulty of the question. We add a classification module to make our approach more adaptive of different question categories. Dense Phrases might be able to provide sufficient signal to compute valid evidence chain. Also, to the best of our understanding, none of the existing approaches have tried to leverage dense phrase representations to answer multihop questions.

### 1.3 Proposed Approach

Current models train to retrieve passages/documents that have high coarse relevance with the question, and then evaluate on the model's ability to predict the fine-grained final answer. We think this presents a mismatch in train-test objectives. We, therefore, propose to relax this constraint and instead allow the model to retrieve paths of relevant phrases that are sufficient to lead to the final answer.

We plan to use data from multihop environment in order to train our module. Our training approach relies on dense phrases instead of the full passage retrieval approach and the entire pipeline can further be subdivided into three categories: (1) Data Homogenization Step (2) Retrieval and Query updation and (3) Classification sub-module. We depart from the traditional retriever-reader paradigm. The explanation for these three individual modules can be found below:

#### 1.3.1 Data Homogenization Step

(Oguz et al., 2020) propose a unifying approach where they homogenize the structures and semi-structured data format like knowledge bases, tables and lists to reduce them into an unstructured textual format. They demonstrate a significant gain in performance by this unifying approach. For knowledge bases, they take data from **Wikidata**, **Freebase** and **DBPedia** and use tabular data from Wikidata Relations (with qualifiers) and Freebase relations (with Compound Value Type entities). Once these data sources have been converted to textual format, we can leverage the Dense Phrases (Lee et al., 2020) architecture to obtain embeddings and index them for our future use. We believe this step would help in getting a performance boost. Since this step is modular, we plan to integrate this towards the end of our approach.

#### 1.3.2 Retrieval and Query updation

One of the pre-training steps before this module at the start of our project would be to build a phrase based indexing. We plan to leverage either Dense Phrases or QA-Infused Pre-training of General-Purpose Contextualized Representations i.e. QUIP (Jia et al., 2021) to generate phrase embeddings and index it for performing maximum inner product search between query and phrase embedded vectors.

As a first step in this module, we embed our query and use indexes of phrases to retrieve top-k phrases. Individual candidate scores are computed using a dot-product operation between the query (or query-evidence after first hop) and phrase vectors. Query-evidence tokens are generated by con-

catenating the question and the evidence phrase tokens separated by [SEP] or [CONT] tokens. We train the query encoder using in-batch and hard negatives (in-passage/in-document) to maximize the probability of the gold answer/evidence phrase.

#### 1.3.3 Classification Module

We build a 4-class classification module on top of the query side finetuning module. These 4 classes include: (a) Yes (b) No (c) Span prediction or (d) NO ANSWER prediction. If we arrive at a prediction that is indicative of "NO ANSWER", we iterate and append the best evidence for further query-evidence construction in an iterative fashion. Then we jointly train the adaptivity module (classification layer) to predict if more hops are required or not.

### 1.4 Likely Challenges and Mitigation

In case we find out from the experiments that the extracted phrases lack surrounding context and are insufficient to answer multihop questions in several cases, we can retrieve top K documents/passages using Dense Phrases (Lee et al., 2021) and generate contextual embeddings of these phrases (concatenated with query or query-evidence tokens) using a pretrained language model which would be then fed into a Fusion In Decoder for further interaction. The fused representations are then fed into the Reader Module to determine whether the answer exists in them which happens by comparing start and end token softmax probabilities with the probability of answer not being there. If so, the retrieval process ends and the QA system delivers the answer span extracted by the Reader Module as the predicted answer. Otherwise, the retrieval process continues to the next hop. The query updation/formation for further iterations happen in the same way as described in the approach section above where the query generator generates the clue span $C$ and concatenate $C$ with the original question $q$ as $C = U(q, D_k)$ and $q = concatenate(q, [SEP], C)$.

## 2 Related Work

A significant work has been done in the Open Domain Question Answering (ODQA) which involve Retriever Reader architectures or just a Dense Phrase(Lee et al., 2020) retrieval to extract the most relevant answer phrase. There are several recent papers like (Zhang et al., 2021) and (Xiong et al., 2020) which try to handle multi hop question answering through iterative document retrieval but differ in the way they retrieve and update query for every iteration/hop. (Zhang et al., 2021) retrieves top documents with most lexical overlap with the question using TF-IDF and then uses a graph based reranking model to maintain an exclusive list of most relevant documents which are then concatenated fed into a Fusion module for further interaction. The reader module then uses these fused representations to determine if we need more hops. Multihop Dense Passage Retrieval approach by (Xiong et al., 2020) iteratively encodes the question and previously retrieved documents as a query vector and retrieves the next relevant documents using efficient MIPS methods. But they do the iterative retrieval statically for 2 hops and do not adapt to the question complexity.

Our approach uses Dense Phrases (Lee et al., 2020) for phrase retrieval and as we are directly retrieving top K phrases from the indexed corpus using Maximum Inner Product Search (MIPS) so there is no need for a reranking module and as there are no documents we do not need a decoder or a reader to do some sort of cross attention or information exchange between documents to answer the question. Our iterative document retrieval and query updation is inspired from (Xiong et al., 2020) where we replace their passage retrieval with the Dense Phrase retrieval approach. Our adaptivity or classification module which determines if we need to take more hops in order to answer the question or not is inspired from (Qi et al., 2020) where they train a classifier which is conditioned on the Transformer encoder representation of the [CLS] token to predict one of the 4 classes SPAN/YES/NO/NOANSWER. Further hops are considered depending on the predicted class. The classifier assigns an *answerability* score to assess the liklihood of the document having the answer to the original question on this reasoning path. The answer phrase is predicted from the context using a span start and a span end classfier.

In our approach we are also aiming to incorporate data homogenization step from Wiki Tables and knowledge bases like Freebase to improve the retrieval accuracy like (Oguz et al., 2020).

## 3 Experiments

### 3.1 Datasets

The datasets that we plan to use can be categorized based on difficulty of the questions. For single hop, we plan to use Natural Questions (Kwiatkowski et al., 2019) and SQuAD (Rajpurkar et al., 2016) open dataset. For evaluating multihop settings, we would primarily work on evaluation using dev set from HotpotQA and also evaluate our model on BeerQA dataset (Qi et al., 2020) for evaluating 3+ hop generalization. The dense phrase representation that we plan to compare as part of the baseline is available on their GitHub repository.

### 3.2 Baselines

For baseline model, we plan to make a comparison with Dense Phrases (Lee et al., 2020) (for single hop phrase retrieval). For the retriever-reader comparisons, we will make a comparison with Multi-hop Dense Retrieval (MDR) approach by (Xiong et al., 2020).

We have publised results from Dense Phrases on Natural Questions dataset for single hop phrase retrieval and question answering and we also have MDR model results on Hotpot QA dataset to compare with. Both these papers have publicly available code which we can use to verify baselines and also to build upon for our approach.

### 3.3 Software

We plan to use PyTorch framework for our implementation of the project. In addition to this, we will use Dense Phrases implementation from (https:/github.comprinceton-nlpDensePhrases) repository for using Phrase based representation. We build other components on our own. In addition to this, we use Unity cluster for our experiments.

### 3.4 Timeline

Our high level timeline till midpoint presentation looks like this:

- **Week 1**: Setup boilerplate codebase (DensePhrases) and get datasets
- **Week 2**: Run single-hop evaluation on multiple datasets
- **Week 3-4**: Implement iterative retrieval for multi-hop
- **Week 5**: Run experiments and evaluate contingencies

We are going to divide implementing baselines and running experiments on different datasets equally among ourselves. Each of us will conduct a different experiment & compare results while working in parallel on the project.

### 3.5 Challenges

The Unik-QA authors (Oguz et al., 2020) have not made their code or the data public so even if data homogenization sounds like a simple step, it might require a significant amount of engineering efforts. Also, in our present formulation, we plan to leverage phrase based embeddings and indexing them for fast retrieval. So it might be the case that we will not have enough surrounding context to answer the question as well as in learning the new query representation for the next-hop retrieval. Also, the main problem in answering multi-hop open-domain questions is that the search space grows exponentially with each retrieval hop, so in order to test whether our approach works well for more than two hop settings, we plan to test our implementation on newly devised BeerQA dataset (Qi et al., 2020).

### 3.6 Contingency Plan

As our approach is based on intuition and prior research results, we have also thought of few alternate approaches & have received positive feedback for those approaches from our mentors. One of them is mentioned at the end of the section 1.4. Depending on results we get after implementing our current approach and our analysis, we either refine our existing approach or we pursue one of our alternate approaches which shares commonalities with current approach (uses phrase based retrieval) but is more inclined towards using a passage retrieval mechanism along with a reader.

## 4 Acknowledgements

We thank our mentors Manzil Zaheer & Dung Thai for useful guidance and also acknowledge the helpful feedback provided by Prof. Mohit Iyyer.

## References

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented

language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938. PMLR.

Robin Jia, Mike Lewis, and Luke Zettlemoyer. 2021. Question answering infused pre-training of general-purpose contextualized representations. *arXiv preprint arXiv:2106.08190*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2020. Learning dense representations of phrases at scale. *arXiv preprint arXiv:2012.12624*.

Jinhyuk Lee, Alexander Wettig, and Danqi Chen. 2021. Phrase retrieval learns passage retrieval, too. *arXiv preprint arXiv:2109.08133*.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2020. Unik-qa: Unified representations of structured and unstructured knowledge for open-domain question answering. *arXiv preprint arXiv:2012.14610*.

Peng Qi, Haejun Lee, Oghenetegiri Sido, Christopher D Manning, et al. 2020. Answering open-domain questions of varying reasoning steps from text. *arXiv preprint arXiv:2010.12527*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

Minjoon Seo, Tom Kwiatkowski, Ankur P Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2018. Phrase-indexed question answering: A new challenge for scalable document comprehension. *arXiv preprint arXiv:1804.07726*.

Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen-tau Yih, Sebastian Riedel, Douwe Kiela, et al. 2020. Answering complex open-domain questions with multi-hop dense retrieval. *arXiv preprint arXiv:2009.12756*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Yuyu Zhang, Ping Nie, Arun Ramamurthy, and Le Song. 2021. Answering any-hop open-domain questions with iterative document reranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 481–490.